8В

УДК 001.51:004.81

А.А. Воронов
Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь
sash_v_oo@tut.by

Метод покрытия прямоугольниками объектов топологии микросхем, основанный на использовании обобщенной диаграммы Вороного

Рассмотрена проблема покрытия многоугольников прямоугольниками, которая имеет место при подготовке входной информации для устройств, выполняющих изготовление фотошаблонов. Входная информация представляет собой описание последовательности прямоугольников. Выбор этой последовательности во многом определяет производительность этих устройств и качество получаемых фотошаблонов. Прямоугольники должны лежать полностью внутри многоугольника, и число их должно быть минимальным или близким к минимальному. Предложен простой эвристический алгоритм, основанный на использовании диаграммы Вороного, который покрывает многоугольник без дыр с острыми внутренними углами при помощи прямоугольников.

Введение

При производстве интегральных схем, фотоэлектрических преобразователей, ЖК-индикаторов, а также многих других микроэлектронных устройств возникает задача формирования топологических структур на металлизированных фотошаблонах [1]. Эти структуры формируются с помощью специальных генераторов изображений. Генераторы изображений строят топологию на фотошаблоне из наборных элементов. Наборный элемент представляет собой прямоугольник. Создание посредством таких генераторов произвольных изображений топологических структур требует предварительного разложения (декомпозиции) этих структур на множество прямоугольников, объединение которых с заданной точностью совпадает с описанием соответствующих исходных структур и формирует покрытие. При этом число прямоугольников, входящих во множество, должно быть минимальным или близким к минимальному. На основании полученного множества прямоугольников формируется оптимальная входная последовательность, которая кодируется в соответствии с правилами входного языка соответствующего генератора изображений. Задачи декомпозиции и, как частный случай, покрытия не являются новыми. Разработаны эффективные методы решения этих задач [1-9], однако при проектных нормах меньше 1 мкм возникают значительные трудности при генерации изображений, так как возрастает сложность топологии.

Известные методы декомпозиции многоугольников можно разбить на переборные [3-5] и сканирующие [1], [6-8]. Переборные методы просты в реализации, но оказываются эффективными лишь при покрытии односвязных и небольших по числу угловых точек контуров многосвязных областей. Методы сканирующего типа [10] являются эффективными как для односвязных, так и для многосвязных областей.

Эти методы дают возможность покрывать области как позитивного, так и негативного фотошаблона. Также используются комбинации этих методов [11]. Перечисленные выше методы являются приближенными, т.е. дают возможность получать решения, близкие к минимальному.

В настоящей работе рассматривается оригинальный приближенный метод декомпозиции многоугольника в совокупность прямоугольников. Решение задачи декомпозиции осуществляется с использованием одной из модификаций диаграммы Вороного, называемой скелетом многоугольника [12]. Скелет многоугольника получается из обобщенной диаграммы Вороного заменой параболических ребер хордами. Декомпозиция многоугольника начинается с его разбиения на ячейки по найденному скелету с целью покрыть каждую из этих ячеек прямоугольником. Если какая-либо из этих ячеек не покрывается прямоугольником, то она делится на более мелкие ячейки и т.д. Процесс декомпозиции завершается после того, как каждая из найденных ячеек будет покрыта прямоугольником.

Метод декомпозиции

Для решения задачи декомпозиции используются как специальные, так и универсальные алгоритмы. Специальные алгоритмы решают задачу покрытия окружностей, колец, шин, треугольников. Универсальный алгоритм решает задачу покрытия произвольного многоугольника.

Универсальный алгоритм декомпозиции является эвристическим. Ниже приводится описание эвристического метода покрытия, являющегося развитием метода, описанного в работе [13]. Метод основан на использовании модификации обобщенной диаграммы Вороного, называемой скелетом многоугольника, и позволяет покрывать произвольные односвязные многоугольники. В процессе декомпозиции исходный многоугольник разбивается на всё меньшие и меньшие части, которые обрабатываются независимо до тех пор, пока каждая из этих частей не будет покрыта прямоугольниками, находящимися в пределах исходного многоугольника. Первое разделение достигается с помощью генерации скелета многоугольника, также называемого «средней осью».

В отличие от диаграммы Вороного, которая строится для множества точек, скелет многоугольника (средняя ось) задает местоположение всех центров кругов, содержащихся в многоугольнике, которые касаются его границы в двух или более точках. Особенностью предлагаемого метода по сравнению с методом из работы [13] является то, что он позволяет находить покрытия для произвольных односвязных многоугольников, в том числе и с острыми внутренними углами. Для покрытия острых углов многоугольника применяется специальный алгоритм, который дополняет основной метод.

Предлагаемый ниже метод декомпозиции состоит из следующих этапов:

1) формирование скелета исходного многоугольника;

2) разбиение многоугольника по его скелету на ячейки;

3) покрытие ячеек прямоугольниками.

Последний этап метода выполняется для каждой ячейки, независимо от результата покрытия других ячеек. Каждая ячейка покрывается за время, линейное относительно числа ребер, включенных в нее. Последовательно рассмотрим каждый этап:

1) формирование скелета для односвязного многоугольника. Рассмотрим задачу построения скелета для многоугольника.

Среди алгоритмов построения диаграммы Вороного для множества отрезков [14-16] следует выделить два алгоритма: Форчуна [15] и Ли [16]. Алгоритм Форчуна весьма сложен для реализации. Кроме того, в силу универсальности, он строит диаграмму Вороного не только внутри, но и вне многоугольника, что является для декомпозиции лишней работой. Более простой (и хронологически более ранний) алгоритм Ли строит диаграмму Вороного внутри многоугольника. Он основан на алгоритмической парадигме «разделяй и властвуй». Исходный многоугольник разбивается на две разомкнутые ломаные линии, для каждой из них рекурсивно строится диаграмма Вороного, а затем осуществляется объединение обеих диаграмм. На практике алгоритм Ли широко используется и работает достаточно надежно. Используемый ниже алгоритм формирования скелета многоугольника является модификацией алгоритма Ли [12]. Скелет используется для исследования топологических и метрических свойств области, что находит применение во многих методах анализа и распознавания изображений [17], [18]. Оценка вычислительной сложности задачи построения скелета многоугольника равна O(n), где n – число вершин многоугольника [7].

Для выполнения первого этапа метода декомпозиции используется диалоговая программа «Скелетизация многоугольной фигуры» [12].

Односвязный многоугольник – это замкнутая ограниченная область в евклидовой плоскости, границу которой составляет замкнутый контур, многоугольник без «дыр».

Скелетом замкнутой области называется геометрическое место точек области, являющихся центрами максимальных по включению вписанных окружностей.

Пример 1. На рис. 1 приведен пример односвязного многоугольника, введенного и визуализированного средствами программы «Скелетизация многоугольной фигуры».

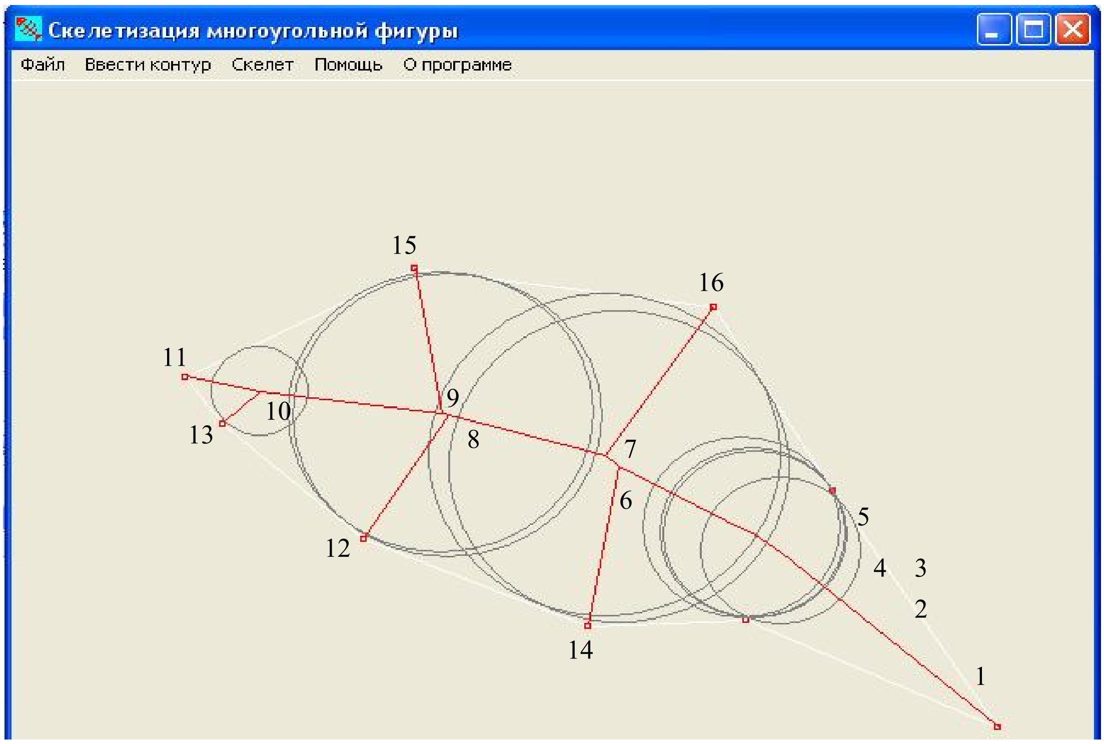


Рисунок 1 – Пример работы модуля реализующего первый шаг алгоритма для односвязного многоугольника

2) разбиение многоугольника на ячейки по его скелету. На первом шаге данного этапа проводим анализ исходного многоугольника на наличие острых внутренних углов. Градусную меру вершины многоугольника определяем как величину угла, образованного прямыми, проходящими через эту вершину по сторонам, примыкающим к ней.

Для острых углов выполняются следующие действия: восстановим перпендикуляры из вершины скелета, ближайшей к вершине с острым углом, на стороны, образующие острый угол. Одновременно эти перпендикуляры являются радиусами окружности, а стороны, к которым они проведены, принадлежат прямым, касательным к ней. Указанные перпендикуляры отсекают от исходного многоугольника четырехугольник с острым углом.

Полученный четырехугольник обладает следующими свойствами.

Стороны-перпендикуляры равны между собой как радиусы окружности. Стороны четырехугольника, примыкающие к перпендикулярам, также равны.

Исходный многоугольник M разбивается на остроугольный четырехугольник и многоугольник M_1, у которого, по сравнению с многоугольником M, на один внутренний острый угол меньше. Повторяем эту процедуру и для многоугольника M_1, если у него имеется хотя бы один внутренний острый угол и т.д. Процесс завершается после получения многоугольника M_n, не содержащего внутренних острых углов. При этом будут получены n четырехугольников с острыми углами.

Пример 2. Для рассмотренного выше в примере 1 скелета указанные преобразования применяются для вершин 1 и 11, так как углы при этих вершинах острые. В результате получим многоугольник, представленный на рис. 2.

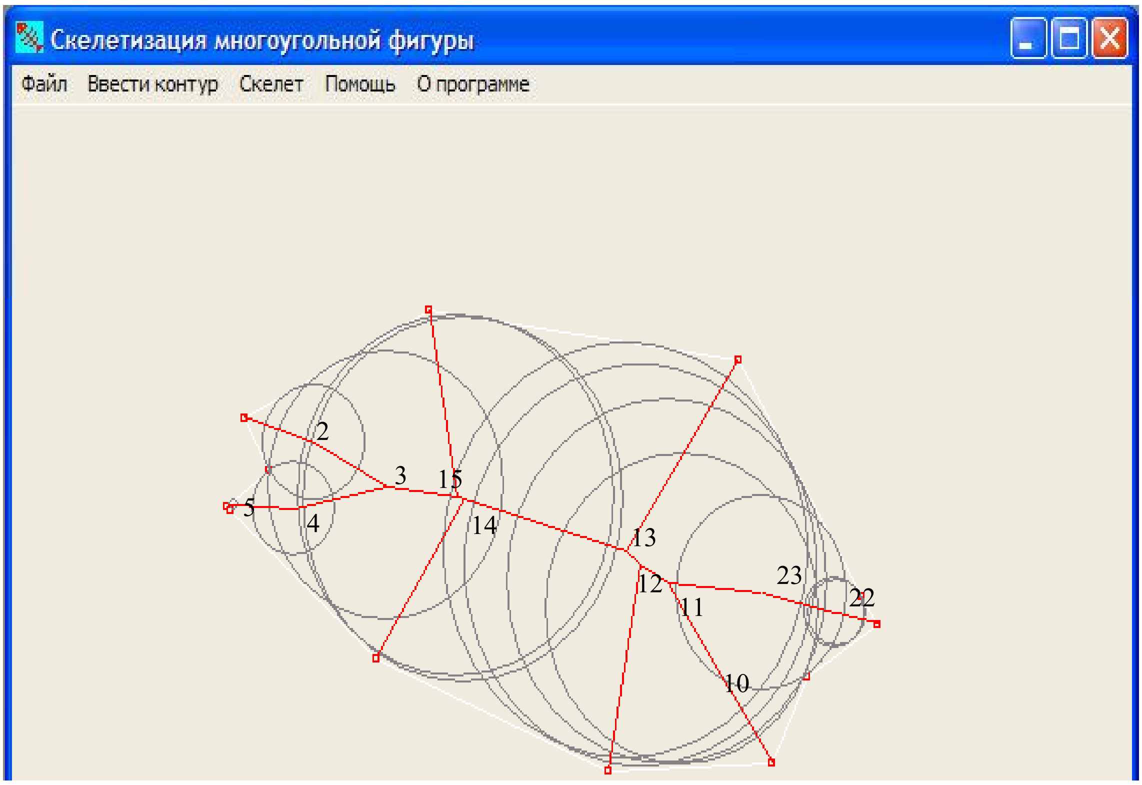


Рисунок 2 – Пример скелета многоугольника
с исключенными острыми углами

Четырехугольник с острым углом покрываем при помощи алгоритма, описанного в специальном подразделе. Для сформированного многоугольника M_n без острых углов повторяем построение скелета и эффективной части скелета (часть скелета, не содержащая вершин, которые лежат на границе многоугольника).

Ячейки для покрытия формируются по сторонам многоугольника или их отрезкам.

Из каждой вершины эффективного скелета опускаем перпендикуляр на прямую, содержащую сторону многоугольника, и затем определяем, принадлежит ли проекция точки данной стороне-отрезку. Если проекция принадлежит стороне-отрезку, то переходим к следующей вершине скелета, в противном случае – проецируем вершину скелета на прямую, содержащую следующую по порядку (обход против часовой стрелки) сторону многоугольника.

Проверка принадлежности точки $A(x,y)$ отрезку z с вершинами $B(x_1,y_1)$ и $C(x_2,y_2)$ выполняется следующим способом:

точки отрезка z можно описать уравнением $pOB + (1-p)OC = z$, $0 <= p <= 1$, OB и OC – векторы. Если существует такое p, $0 <= p <= 1$, что $pOB + (1-p)OC = A$, то A лежит на отрезке, иначе – нет. Распишем в виде системы уравнений:

$$px_1 + (1-p)x_2 = x$$

$$py_1 + (1-p)y_2 = y.$$

Из первого уравнения находим p, подставляем во второе: если получаем равенство и $0 <= p <= 1$, то A на отрезке, иначе – нет.

Пусть g будет отрезком многоугольника P, который порождает ячейку. В ячейке может быть только три или четыре вершины, опускаем из каждой вершины эффективного скелета перпендикуляр к соответствующей стороне многоугольника P. Назовем сторону каждой ячейки, которая находится на стороне многоугольника P, основой ячейки. Отрезки, которые перпендикулярны ее основе, – сторонами ячейки и границу ячейки, противолежащую ее основе, – вершиной ячейки. Ячейка является тривиально покрываемой наименьшим прямоугольником тогда и только тогда, когда одна сторона прямоугольника – основа ячейки, и прямоугольник покрывает ячейку таким образом, что сам полностью лежит в пределах многоугольника.

Пример 3. Результат построения ячеек по эффективному скелету для рассматриваемого многоугольника из примера 2 приведен на рис. 3.

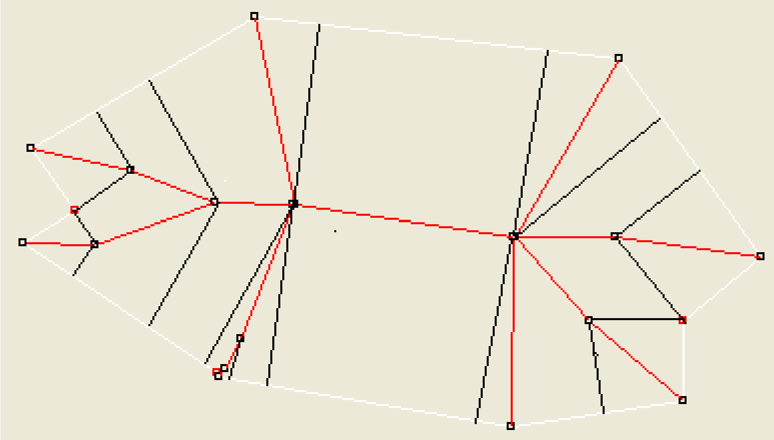


Рисунок 3 – Примеры формирования ячеек для многоугольника

3) покрытие ячеек при помощи прямоугольников.

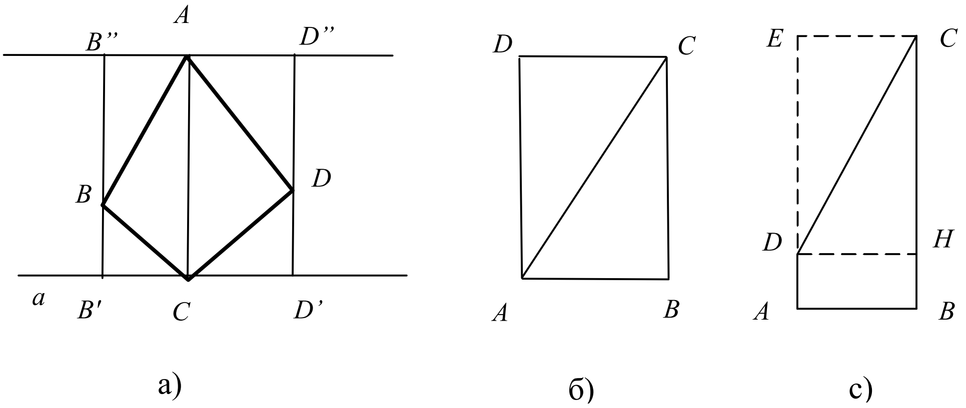


Рисунок 4 – Примеры покрытия: а) покрытие ячейки порожденной вершиной; б) покрытие ячейки треугольника; c) ячейка – трапеция, угол (*A*, *D*, *C*) – по крайней мере 135°

После исключения острых углов не все ячейки порождаются отрезками сторон многоугольника, потому что в получившемся многоугольнике некоторые углы больше 180°, в исходном многоугольнике также возможны такие случаи. Так на рис. 4 а) изображена такая ячейка: вершина *A* соответствует вершине скелета 3 на рис. 2, вершина *B* соответствует вершине скелета 2 на рис. 2, вершина *D* соответствует вершине скелета 4 на рис. 2, а вершина *C* соответствует вогнутой вершине многоугольника (угол при которой больше 180°) на рис. 2. Проводим прямую *AC*, через *C* проводим прямую *a*, перпендикулярную *AC*. Через *A* также проводим прямую, перпендикулярную *AC*. Опускаем из *B* и *D* перпендикуляры на последние две построенные прямые. Так находим координаты вершин прямоугольника покрытия *B'D'D"B"*.

Каждая ячейка обрабатывается независимо. Действие алгоритма описано отдельно для различных типов ячеек.

Ячейка-треугольник.

Треугольник *ABC* полностью покрывается при помощи четырехугольника *ABCD* на рис. 4 б), так как угол при *B* прямой (*CB* перпендикуляр к *AB*).

Прямоугольник покрытия находится полностью в пределах многоугольника *P*, так как после исключения острых углов в многоугольнике нет острых углов и значит ячейки, которые имеют только три вершины, имеют тривиальное покрытие.

Ячейка-трапеция

Если ячейка – трапеция, тогда она является тривиально покрываемой. Иначе, пусть *A, B, C* и *D* будут вершинами ячейки в порядке против часовой стрелки, где *A* и *B* – конечные точки ее отрезка-основы. Пусть $|AD| < |BC|$ и основа горизонтальна и находится снизу. Теперь, если угол (*A, D, C*) больше или равен 135°, то мы утверждаем, что ячейка – тривиально покрываема. Пусть *ABCE* будет прямоугольник, который покрывает ячейку, рис. 4 с). Его часть *ABHD* – это прямоугольник и находится в пределах ячейки. Следовательно, он находится также в пределах многоугольника.

Прямоугольник *DHCE* – это покрытие треугольника *DHC* и находится внутри многоугольника, аналогично ячейке-треугольнику. Теперь нам остался один случай, когда угол (*A, D, C*) больше чем 90° и меньше чем 135°. Оценивая максимальную высоту любого прямоугольника в пределах многоугольника с основой *AB*, можно сказать, что она будет $> |BC|$, повернув ячейку вокруг ее вершины (с конечными точками *C* и *D*), получим симметрическое изображение ячейки относительно ее вер-

шины, рис. 5. Пусть *A′* и *B′* будут симметрическим изображением *A* и *B* соответственно. Пусть *E* будет точкой на *A′B′* или на *B′C*, такой, что отрезок *AE* проходит через нее. Если $|AE| > |BC|$, то ячейка тривиально покрываема. Еще остается рассмотреть случай $|AE| < |BC|$. Сформулируем следующее определение.

Если $|AE| < |BC|$, то ячейка *ABCD* называется ячейкой-тоннелем для многоугольника *P*. Многоугольник, определенный вершинами *ABCB′A′* (рис. 5), следующими в порядке против часовой стрелки, называется тоннелем.

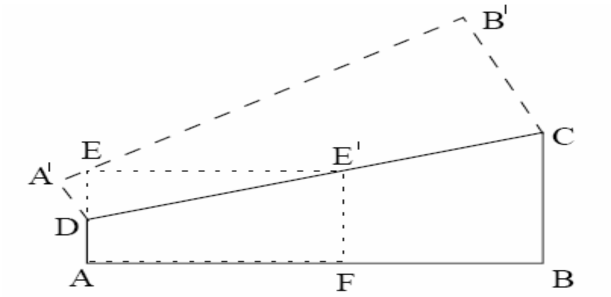


Рисунок 5 – Пример образования ячейки-тоннеля

Если $|AE| < |BC|$, то *E* находится на *A′B′*. Пусть α будет угол (*A, D, C*) минус 90°. Мы получаем, что угол (*A′, D, E*) = 2α. Кроме того, мы вычисляем $|DE| = |A'D|/\cos(A', D, E)$. Так как $|A'D| = |AD|$, то из этого следует, что $|AE| = |AD| + |DE| = |AD|*(1+1/\cos 2\alpha)$. Определяем коэффициент тоннеля как $(1+1/\cos 2\alpha)$. Теперь алгоритм разбивает ячейку на две подъячейки, *AFE′D* и *FBCE′*, где *EE′* является параллельной *AB*, и *E′* лежит на *DC*. Подъячейка *AFE′D* покрыта тривиально. Если подъячейка *FBCE′* не является тривиально покрываемой, то обрабатывается таким же образом, как ячейка-тоннель. Таким образом, первоначальная ячейка *ABCD* в конечном счете разделена на $[(\lg(|BC/AD|))/\lg c]$ тривиально покрываемые подъячейки, где с – коэффициент тоннеля.

Исходя из вышесказанного, мы можем сказать, что после построения скелета, за время *O(n)*, любой многоугольник без дыр *P*, кроме его ячеек-тоннелей, может быть покрыт *O(n)* прямоугольниками за время *O(n)*. В пределах того же времени все ячейки-тоннели в *P*, если они есть, могут быть обнаружены. В частности может быть *O(n)* ячеек-тоннелей в *P*.

Пример 4. Результат покрытия всех ячеек для многоугольника без острых углов из примера 3 приведен на рис. 6.

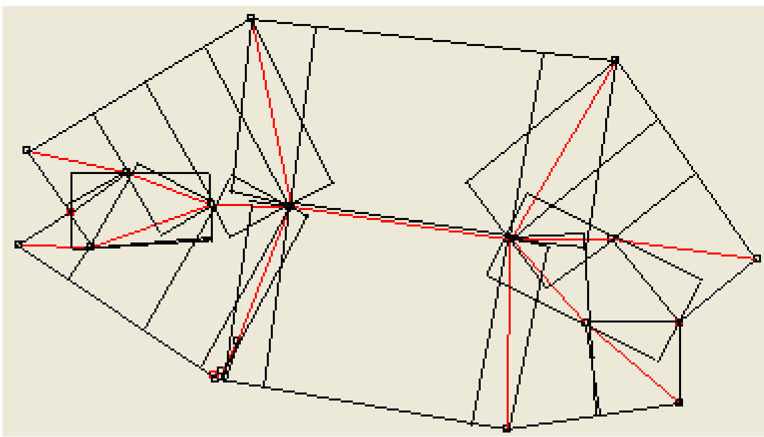


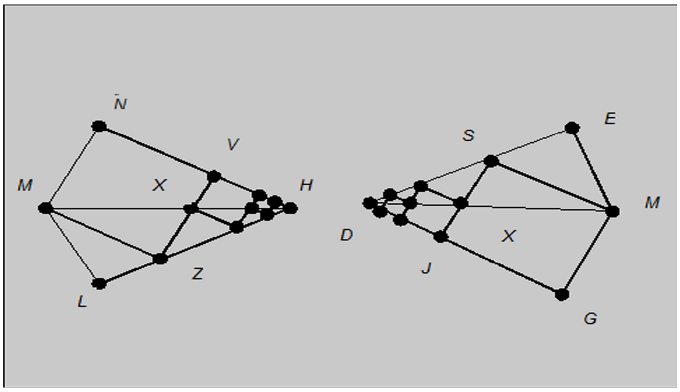
Рисунок 6 – Пример покрытия прямоугольниками многоугольника с исключенными острыми углами

Покрытие четырехугольника с острым углом при помощи прямоугольников

Для четырехугольника с острым углом используется следующий алгоритм его покрытия прямоугольниками.

Производим построение первого прямоугольника, две вершины которого лежат на одной стороне острого угла четырехугольника – вершины N и V, третья – на другой стороне – вершина Z, а четвертая – на биссектрисе острого угла (ребро скелета) – вершина M (рис. 7 а). Для следующего прямоугольника одной из вершин будет точка X – точка пересечения стороны построенного прямоугольника с биссектрисой острого угла. Найдя ее, производим далее построение прямоугольника.

Строим прямоугольники, исходя из указанной последовательности действий, на одной стороне острого угла четырехугольника до тех пор, пока меньшая сторона прямоугольника не станет меньше, чем величина d – минимально допустимый размер, который может позиционировать лазерный генератор. Производим точно такое же построение прямоугольников на другой стороне угла (рис. 7 б).



а) б)

Рисунок 7 – Пример четырехугольника с острым углом: а) б) – покрытие его

Оценка искажения, получаемого в рассмотренном выше алгоритме покрытия острого угла, равна величине $d^2\text{ctg}(0{,}5\alpha)$, где d – минимально допустимый размер, который может позиционировать лазерный генератор, α – величина острого угла [19]. С учетом технологии изготовления интегральных микросхем это искажение еще меньше. Острые углы, как правило, не функциональны, и подобная аппроксимация в них вполне допустима, а иногда даже и полезна [19].

Заключение

Предлагаемый алгоритм решения задачи декомпозиции позволяет находить покрытие более высокого качества, так как за счет предварительной скелетизации многоугольника в результирующем покрытии минимизирована площадь взаимных пересечений прямоугольников, что снижает вероятность переэкспонирования резиста и, как следствие, его вздутия и выбраковки фотошаблонов. Следовательно, предлагаемый алгоритм дает возможность формировать более качественные входные последовательности для генераторов изображений.

Литература

1. Фейнберг В.З. Геометрические задачи машинной графики больших интегральных схем / В.З. Фейнберг. – М. : Радио и связь 1987. – 178 с.
2. Ferrari L. Minimal rectangular partitions of digitized blobs. Computer Vision / L. Ferrari, P.V. Sankar, J. Sklansky // Graphics, and Image Processing. – 1984. – № 28. – P. 58-71.
3. Забродская В.П. Машинное проектирование фотошаблонов для телевизионных ФЭП / В.П. Забродская, Б.А. Котов, И.С. Серпов // Электронная промышленность. – 1974. – № 3. – С. 56- 59.
4. Казеннов Г.Г. Алгоритм подготовки данных для микрофотонаборных установок / Г.Г. Казеннов, Л.Б. Осипов, В.М. Щемелинин, А.Л. Стемпковский // Электронная промышленность. – 1974. – № 6. – С. 84-87
5. Носова Е.Г. Алгоритмы разбиения плоских фигур в системах машинного проектирования интегральных схем / Е.Г. Носова, А. Г. Свердлов, В.З. Фейнберг // Изв. АН БССР. Сер. физ.- мат. наук. – 1978. – № 5. – С. 16-23.
6. Осипов Л.Б. Программа фотонабора топологий, содержащих многосвязные фигуры и наклонные линии // Автоматизация РЭА и ЭВА. – Пенза, 1977. – С. 50-53.
7. Осипов Л.Б. Алгоритмические методы подготовки и контроля информации для микрофотонаборных установок / Л.Б. Осипов, Г.Э. Широ // Электронная техника. Сер. 10. Микроэлектронные устройства. – 1978. – Вып. 2. – С. 89-103.
8. Стемпковский А.Л. Универсальный алгоритм подготовки данных для микрофотонаборных установок / А.Л. Стемпковский // Электронная техника. Сер. 3. Микроэлектронника. – 1978. – Вып. 6. – С. 74-80.
9. Ohtsuki T. Minimum dissection of rectilinear regions / T. Ohtsuki // In Proceedings of the 1982 IEEE International Symposium on Circuits and Systems. – Rome, 1982. – P. 1210-1213.
10. Hegedus A. Algorithms for covering polygons by rectangles / A. Hegedus // Computer Aided Design. – Vol. 14, № 5. – 1982.
11. Asano Ta. Partitioning a polygonal region into trapezoids / Ta Asano, Te. Asano, H. Imai // J. ACM. – 1986. – № 33. – P. 290-312.
12. Местецкий Л.М. Скелетизация многоугольной фигуры на основе обобщенной триангуляции Делоне / Л.М. Местецкий // Программирование. – №. 3. – 1999. – С. 16-31.
13. GudmundssonJ. Close approximations of minimum rectangular coverings // J. Gudmundsson, C. Levcopoulos // In FST & TCS'96, volume 1180 of LNCS. – 1996. – P. 135-146.
14. Kirkpatrick D.G. Efficient computation of continuous skeletons / D.G. Kirkpatrick // In Proceedings of the 20th Annual IEEE Symposium on FOCS. – 1979. – P. 18-27.
15. Fortune S. A sweepline algorithm for Voronoi diagrams / S. Fortune // Algorithmica. – 1987. – № 2. – P. 153-174.
16. Lee D.T. Medial axes transform of planar shape / D.T. Lee // IEEE Trans. Patt. Anal. Mach. Intell. PAMI-4. – 1982. – P. 363-369.
17. Препарата Ф. Вычислительная геометрия: введение / Ф. Препарата, М. Шеймос. – Москва : Мир, 1989.
18. Pavlidis T. Algorithms for Graphics and Image Processing / T. Pavlidis // Computer Science Press, Rockville, MD. – 1982.
19. Техническое описание и инструкция по эксплуатации генератора изображений ЭМ-5109. – Минск : КБТЕМ-ОМО, 2001. – 323 с.

А.А. Воронов

Метод покриття прямокутниками об'єктів топології мікросхем, що грунтується на використанні узагальненої діаграми Вороного

Розглянуто проблему покриття багатокутників, що виникає під час підготовки вхідної інформації для приладів, які виконують виготовлення фотошаблонів. Вхідна інформація являє собою опис послідовності прямокутників. Вибір цієї послідовності більшою мірою визначає продуктивність цих приладів і якість отримуваних фотошаблонів. Прямокутники повинні знаходитися повністю в середині багатокутника, і кількість їх повинна бути мінімальною або близькою до мінімальної. Запропонований простий евристичний алгоритм, що грунтується на використанні діаграми Вороного, який покриває багатокутник без дірок з гострими внутрішніми кутами за допомогою прямокутників.

A.A. Voronov

Method for Covering of IC Layout Patterns by Rectangles Based on Voronoi's Diagram

The problem of covering polygons by rectangles that take place in input data preparation for integrated circuit layout generators is considered. Input data is the sequence of rectangles. Basically this sequence determines the productivity of these integrated circuit layout generator and quality of output photomask. The rectangles must lie entirely within the polygon and it is preferable to cover the polygon with as few rectangles as possible. The simple heuristic algorithm, based on the Voronoi's diagrams, that cover hole-free polygon with acute interior angles by rectangles is presented.

Статья поступила в редакцию 04.06.2009.